I claim:

1. A method of protecting application program software including

actuating a tracer function to copy $2^{1 \text{ to } n}$ instructions from the API code;

storing and executing said instructions;

returning to the next instruction $(2^{(1 \text{ to } n)+1})$of the API code, wherein $2^{1 \text{ to } n}$ represents the number of instructions and n is the maximum number of instructions describing the API code.

2. A method of protecting application program software including

actuating a tracer function to copy $2^{1 \text{ to } n}$ instructions from the API code;

storing and executing said instructions;

returning to the next instruction $(2^{(1 \text{ to } n)+1})$of the API code, wherein $2^{1 \text{ to } n}$ represents the number of instructions and n is the maximum number of instructions describing the API code, wherein the number of instructions is 16 and the copied instructions are stored in the Random Access Memory (RAM) of the CPU.

3. A method as claimed in claim 1 wherein the application program software is security program software.

4. A method of protecting application program software as claimed in claim 1 wherein the tracer function includes the following instructions:

read instruction of *myfunction* (interpret opcodes);

if instruction is *not* ((a call, jmp, sysenter, syscall or branch instruction which ends up out of scope) or (less than the 16th instruction)) then copy to the local buffer;

repeat above steps until out of scope;

execute the local buffer;

continue execution in the original *myfunction* code at the offset where the out of scope instruction was encountered.